Theoretische Informatik 2

Johannes Köbler

Institut für Informatik
Humboldt-Universität zu Berlin

WS 2009/10

Termine

Vorlesung

Di	09-11	RUD 25	3.001	J. Köbler
Do	09-11	RUD 25	3.001	J. Köbler

Übungen

Di	11-13	RUD 26	1'307	S. Tazari
Di	11-13	RUD 25	3.101	W. Kössler
Di	15-17	RUD 26	1'305	S. Tazari
Do	11-13	RUD 26	1'307	S. Kuhnert
Fr	09-11	RUD 25	3.113	W. Kössler

Weitere Infos unter `www.informatik.hu-berlin.de/forschung/gebiete/algorithmenII/Lehre/ws09/theo2`

Übungen (Anmeldung über GOYA erforderlich)

Ausgabe der Aufgabenblätter

- in der VL, auf GOYA und der VL-Webseite

Bearbeitung

- in Gruppen von maximal drei Teilnehmern
- bitte Namen und Matrikelnr. angeben

Abgabe

- bis 9:10 Uhr im Erdgeschoss von Haus 4!
  (Abgabetermin s. Aufgabenblatt)
- bitte nur in Papierform

Rückgabe

- in den Übungsgruppen

Schein, Klausur, Skript

Scheinkriterien

- Lösen von $\geq 50\%$ der schriftlichen Aufgaben,
- Erfolgreiches Vorrechnen von ≥ 3 mündl. Aufgaben.

Klausur

- **Termin: 23.02.2010**
- Zulassung nur mit Übungsschein
- Nachklausur: nach dem SS 2010

Skript

- wird wöchentlich ins Netz gestellt.

Gibt es zum organisatorischen Ablauf noch Fragen?

Inhalt der Vorlesung

Zentrale Themen von ThI 1:

- Mathem. Grundlagen der Informatik, Beweise führen, Modellierung
  Aussagenlogik, Prädikatenlogik
- Welche Probleme sind lösbar? Berechenbarkeitstheorie

Zentrale Themen von ThI 2:

- **Welche Rechenmodelle sind adäquat? Automatentheorie**
- **Welcher Aufwand ist nötig? Komplexitätstheorie**

Zentrales Thema von ThI 3:

- Wie lassen sich praktisch relevante Problemstellungen möglichst effizient lösen? Algorithmik

Maschinenmodelle

- Rechenmaschinen spielen in der Informatik eine zentrale Rolle.
- Je nach Berechnungskraft unterschiedliche math. Modelle.
- In ThI1: TM als ein universales Berechnungsmodell.
- In ThI3: Registermaschine (engl. random access machine; RAM).
- Dieses Modell ist etwas flexibler als die TM, da es den unmittelbaren Lese- und Schreibzugriff (random access) auf eine beliebige Speichereinheit (Register) erlaubt.
- Hier betrachten wir Einschränkungen des TM-Modells, die vielfältige praktische Anwendungen haben, wie z.B.
    - endliche Automaten (DFA, NFA),
    - Kellerautomaten (PDA, DPDA) etc.

Der Algorithmenbegriff

- Der Begriff Algorithmus geht auf den persischen Gelehrten Muhammed Al Chwarizmi (8./9. Jhd.) zurück.
- Ältester bekannter nicht-trivialer Algorithmus:
  Euklidischer Algorithmus zur Berechnung des größten gemeinsamen Teilers zweier natürlicher Zahlen (300 v. Chr.).
- Von einem Algorithmus wird erwartet, dass er bei jeder zulässigen Problemeingabe nach endlich vielen Rechenschritten eine korrekte Ausgabe liefert.
- Ein Algorithmus ist ein „Verfahren“ zur Lösung eines Berechnungsproblems, das sich prinzipiell auf einer Turingmaschine implementieren lässt (Church-Turing-These).
- Problemeingaben können Zahlen, Formeln, Graphen etc. sein.
- Diese werden über einem Eingabealphabet Σ kodiert.

Alphabet, Wort, Sprache

Definition

- Ein Alphabet ist eine geordnete endliche Menge

  $$\Sigma = \{a_1, \ldots, a_m\}, \ m \geq 1$$

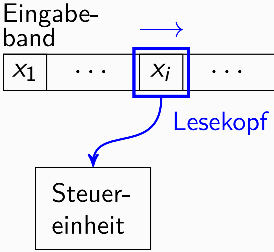
  von Zeichen a_i.
- Eine Folge $x = x_1 \ldots x_n \in \Sigma^n$ heißt Wort (der Länge n).
- Die Menge aller Wörter über Σ ist

  $$\Sigma^* = \bigcup_{n \geq 0} \Sigma^n.$$

- Das (einzige) Wort der Länge $n = 0$ ist das leere Wort, welches wir mit ε bezeichnen.
- Jede Teilmenge $L \subseteq \Sigma^*$ heißt Sprache über dem Alphabet Σ.

Das Rechenmodell des endlichen Automaten

Ein endlicher Automat



- ist eine „abgespeckte“ Turingmaschine,
- verfügt nur über kostant viel Speicherplatz,
- macht bei Eingaben der Länge n nur n Rechenschritte und
- liest in jedem Schritt genau ein Eingabezeichen.

Formale Definition eines endlichen Automaten

Definition

Ein endlicher Automat (kurz: DFA; *deterministic finite automaton*) wird durch ein 5-Tupel $M = (Z, \Sigma, \delta, q_0, E)$ beschrieben, wobei

- $Z \neq \emptyset$ eine endliche Menge von Zuständen,
- Σ das Eingabealphabet,
- $\delta : Z \times \Sigma \to Z$ die Überführungsfunktion,
- $q_0 \in Z$ der Startzustand und
- $E \subseteq Z$ die Menge der Endzustände ist.

Die von M akzeptierte oder erkannte Sprache ist

$$L(M) = \left\{ x_1 \cdots x_n \in \Sigma^* \;\middle|\; \begin{array}{c} \exists\, q_1, \ldots, q_{n-1} \in Z, q_n \in E : \\ \delta(q_i, x_{i+1}) = q_{i+1} \text{ für } i = 0, \ldots, n-1 \end{array} \right\}.$$

DFAs beherrschen Modulare Arithmetik

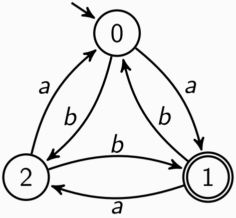
Beispiel

Sei $M_3 = (Z, \Sigma, \delta, 0, E)$ ein DFA mit $Z = \{0, 1, 2\}$, $\Sigma = \{a, b\}$, $E = \{1\}$ und

δ	0	1	2
a	1	2	0
b	2	0	1

◁

Graphische Darstellung:



Der Startzustand wird durch einen Pfeil und Endzustände werden durch einen doppelten Kreis gekennzeichnet.

Behauptung

Die von M_3 erkannte Sprache ist

$$L(M_3) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}, \quad \text{wobei}$$

- $\#_a(x)$ die Anzahl der Vorkommen von a in x bezeichnet und
- $i \equiv_m j$ bedeutet, dass $i - j$ durch m teilbar ist.

Das Erreichbarkeitsproblem für DFAs

Behauptung

Die von M_3 erkannte Sprache ist $\{x \in \{a,b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}$.

Beweis der Behauptung durch Induktion über die Länge von x

Wir betrachten zunächst das Erreichbarkeitsproblem für DFAs.

Frage

Sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA und sei $x = x_1 \cdots x_n \in \Sigma^*$. Welchen Zustand erreicht N bei Eingabe x nach i Schritten?

Antwort

- nach 0 Schritten: q_0,
- nach 1 Schritt: $\delta(q_0, x_1)$,
- nach 2 Schritten: $\delta(\delta(q_0, x_1), x_2)$,
- nach i Schritten: $\delta(\ldots\delta(\delta(q_0, x_1), x_2), \ldots x_i)$.

Das Erreichbarkeitsproblem für DFAs

Definition

- Bezeichne $\hat{\delta}(q, x)$ denjenigen Zustand, in dem sich M nach Lesen von x befindet, wenn M im Zustand q gestartet wird.
- Dann können wir die Funktion

  $$\hat{\delta} : Z \times \Sigma^* \to Z$$

  induktiv (über die Länge von x) wie folgt definieren.
- Für $q \in Z$, $x \in \Sigma^*$ und $a \in \Sigma$ sei

  $$\begin{aligned} \hat{\delta}(q, \varepsilon) &= q, \\ \hat{\delta}(q, xa) &= \delta(\hat{\delta}(q, x), a). \end{aligned}$$

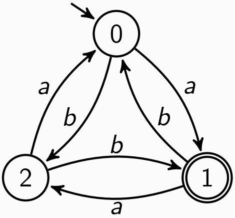
- Die von M erkannte Sprache lässt sich nun auch in der Form

  $$L(M) = \{x \in \Sigma^* \mid \hat{\delta}(q_0, x) \in E\}$$

  schreiben.

DFAs beherrschen Modulare Arithmetik

M_3



Behauptung

$L(M_3) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}$.

Beweis

- 1 ist der einzige Endzustand von M.
- Daher ist $L(M_3) = \{x \in \Sigma^* \mid \hat{\delta}(0, x) = 1\}$.
- Folglich reicht es, folgende Kongruenzgleichung zu zeigen:

$$\hat{\delta}(0, x) \equiv_3 \#_a(x) - \#_b(x)$$

DFAs beherrschen Modulare Arithmetik

Beweis von $\hat{\delta}(0,x) \equiv_3 \#_a(x) - \#_b(x)$:

Wir führen Induktion über die Länge n von x.

Induktionsanfang $n = 0$: klar, da $\hat{\delta}(0,\varepsilon) = \#_a(\varepsilon) = \#_b(\varepsilon) = 0$ ist.

Induktionsschritt $n \rightsquigarrow n+1$:

- Sei $x = x_1 \cdots x_{n+1}$ gegeben und sei $i = \hat{\delta}(0, x_1 \cdots x_n)$.
- Nach IV ist
$$i \equiv_3 \#_a(x_1 \cdots x_n) - \#_b(x_1 \cdots x_n).$$
- Wegen $\delta(i,a) \equiv_3 i+1$ und $\delta(i,b) \equiv_3 i-1$ folgt
$$\delta(i, x_{n+1}) \equiv_3 i + \#_a(x_{n+1}) - \#_b(x_{n+1}) = \#_a(x) - \#_b(x).$$
- Folglich ist
$$\hat{\delta}(0,x) = \delta(\hat{\delta}(0, x_1 \cdots x_n), x_{n+1}) = \delta(i, x_{n+1}) \equiv_3 \#_a(x) - \#_b(x).$$

$\square$

Die Klasse der regulären Sprachen

Definition

Eine von einem DFA akzeptierte Sprache wird als regulär bezeichnet. Die zugehörige Sprachklasse ist

$$\mathrm{REG} = \{L(M) \mid M \text{ ist ein DFA}\}.$$

Frage

Welche Sprachen gehören zu REG und welche nicht?

Singletons sind regulär

Vereinbarung

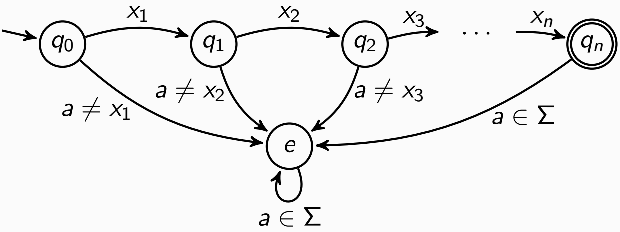
Für das Folgende sei $\Sigma = \{a_1, \ldots, a_m\}$ ein fest gewähltes Alphabet.

Beobachtung 1

Alle Sprachen, die nur ein Wort $x = x_1 \cdots x_n \in \Sigma^*$ enthalten, sind regulär.

Beweis

Folgender DFA M erkennt die Sprache $L(M) = \{x\}$:



□

Abschlusseigenschaften von Sprachklassen

Definition

Ein (k-stelliger) Sprachoperator ist eine Abbildung op, die k Sprachen $L_1, \ldots, L_k$ auf eine Sprache $op(L_1, \ldots, L_k)$ abbildet.

Beispiel

Der 2-stellige Schnittoperator bildet zwei Sprachen L_1 und L_2 auf die Sprache $L_1 \cap L_2$ ab. $\triangleleft$

Definition

- Eine Sprachklasse $\mathcal{K}$ heißt unter op abgeschlossen, wenn gilt:

$$L_1, \ldots, L_k \in \mathcal{K} \Rightarrow op(L_1, \ldots, L_k) \in \mathcal{K}.$$

- Der Abschluss von $\mathcal{K}$ unter op ist die kleinste Sprachklasse $\mathcal{K}'$, die $\mathcal{K}$ enthält und unter op abgeschlossen ist.

Reguläre Sprachen sind unter Komplement abgeschlossen

Beobachtung 2

Ist $L \in \mathrm{REG}$, so ist auch die Sprache $\overline{L} = \Sigma^* \setminus L$ regulär.

Beweis

- Sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA mit $L(M) = L$.
- Dann akzeptiert der DFA

$$\overline{M} = (Z, \Sigma, \delta, q_0, Z \setminus E)$$

das Komplement $\overline{L}$ von L. $\square$

Reguläre Sprachen sind unter Durchschnitt abgeschlossen

Beobachtung 3

Sind $L_1, L_2 \in \mathrm{REG}$, so ist auch die Sprache $L_1 \cap L_2$ regulär.

Beweis

- Seien $M_i = (Z_i, \Sigma, \delta_i, q_i, E_i)$, $i = 1, 2$, DFAs mit $L(M_i) = L_i$.
- Dann wird der Schnitt $L_1 \cap L_2$ von dem DFA

$$M = (Z_1 \times Z_2, \Sigma, \delta, (q_1, q_2), E_1 \times E_2)$$

mit

$$\delta((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$$

erkannt. $\square$

Reguläre Sprachen sind unter Vereinigung abgeschlossen

Beobachtung 4

Die Vereinigung $L_1 \cup L_2$ von regulären Sprachen L_1 und L_2 ist regulär.

Beweis

Es gilt $L_1 \cup L_2 = \overline{(\overline{L_1} \cap \overline{L_2})}$. $\square$

Frage

Wie sieht der zugehörige DFA aus?

Antwort

$$M' = (Z_1 \times Z_2, \Sigma, \delta, (q_1, q_2), (E_1 \times Z_2) \cup (Z_1 \times E_2)).$$

REG ist unter Mengenoperationen abgeschlossen

Korollar

Die Klasse REG der regulären Sprachen ist unter folgenden Operationen abgeschlossen:

- Komplement,
- Durchschnitt,
- Vereinigung.

Wie umfangreich ist REG?

Folgerung

- Aus den Beobachtungen folgt, dass alle endlichen und alle co-endlichen Sprachen regulär sind.
- Da die reguläre Sprache

$$L(M_3) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}$$

  weder endlich noch co-endlich ist, haben wir damit allerdings noch nicht alle regulären Sprachen erfasst.

Konstruktive Charakterisierung von REG

Frage

Lassen sich alle regulären Sprachen aus endlichen Sprachen mithilfe von einfachen Operationen gewinnen?

Definition

- Das Produkt (Verkettung, Konkatenation) der Sprachen L_1 und L_2 ist
$$L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\}.$$
- Ist $L_1 = \{x\}$ einelementig (diese Sprachen werden auch als Singletonsprachen bezeichnet), so schreiben wir für $\{x\}L_2$ auch einfach xL_2.
- Die n-fache Potenz L^n einer Sprache L ist induktiv definiert durch
$$L^n = \begin{cases} \{\varepsilon\}, & n = 0, \\ L^{n-1}L, & n > 0. \end{cases}$$
- Die Sternhülle von L ist $L^* = \bigcup_{n \geq 0} L^n$.
- Die Plushülle von L ist $L^+ = \bigcup_{n \geq 1} L^n = LL^*$.

Überblick

Ziel

Zeige, dass REG als Abschluss der endl. Sprachen unter Vereinigung, Produktbildung und Sternhülle charakterisierbar ist.

Problem

Bei der Konstruktion eines DFA für das Produkt $L_1 L_2$ bereitet es Schwierigkeiten, den richtigen Zeitpunkt für den Übergang von (der Simulation von) M_1 zu M_2 zu finden.

Lösungsidee

Ein nichtdeterministischer Automat (NFA) kann den richtigen Zeitpunkt für den Übergang „erraten“.

Verbleibendes Problem

Zeige, dass auch NFAs nur reguläre Sprachen erkennen.

Nichtdeterministische Automaten

Definition

- Ein nichtdet. endl. Automat (kurz: NFA; *nondet. finite automaton*)

  $$N = (Z, \Sigma, \delta, Q_0, E)$$

  ist genau so aufgebaut wie ein DFA, nur dass er
  - eine Menge $Q_0 \subseteq Z$ von Startzuständen hat und
  - die Überführungsfunktion folgende Form hat:

    $$\delta : Z \times \Sigma \to \mathcal{P}(Z).$$

  - Hierbei bezeichnet $\mathcal{P}(Z)$ die Potenzmenge (also die Menge aller Teilmengen) von Z. Diese wird auch oft mit 2^Z bezeichnet.
- Die von einem NFA N akzeptierte Sprache ist

  $$L(N) = \left\{ x_1 \cdots x_n \in \Sigma^* \,\middle|\, \begin{array}{l} \exists\, q_0 \in Q_0, q_1, \ldots, q_{n-1} \in Z, q_n \in E: \\ q_{i+1} \in \delta(q_i, x_{i+1}) \text{ für } i = 0, \ldots, n-1 \end{array} \right\}.$$

Eigenschaften von NFAs

- Ein NFA N kann also nicht nur eine, sondern mehrere verschiedene Rechnungen parallel ausführen.
- Die Eingabe x wird genau dann akzeptiert, wenn mind. eine Rechnung von N nach Lesen von x einen Endzustand erreicht.
- Im Gegensatz zu einem DFA, der jede Eingabe zu Ende liest, kann ein NFA N „stecken bleiben".
- Dieser Fall tritt ein, wenn N in einen Zustand q gelangt, in dem er das nächste Eingabezeichen x_i wegen

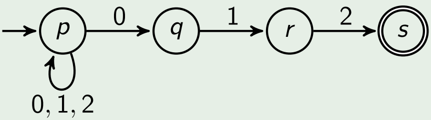
  $$\delta(q, x_i) = \emptyset$$

  nicht verarbeiten kann.

Eigenschaften von NFAs

Beispiel

- Betrachte den NFA N,



d.h. $N = (Z, \Sigma, \delta, Q_0, E)$ mit $Z = \{p, q, r, s\}$, $\Sigma = \{0, 1, 2\}$, $Q_0 = \{p\}$, $E = \{s\}$ und der Überführungsfunktion

δ	p	q	r	s
0	$\{p, q\}$	$\emptyset$	$\emptyset$	$\emptyset$
1	$\{p\}$	$\{r\}$	$\emptyset$	$\emptyset$
2	$\{p\}$	$\emptyset$	$\{s\}$	$\emptyset$

- Dann ist $L(M) = \{x012 \mid x \in \Sigma^*\}$ die Sprache aller Wörter, die mit dem Suffix 012 enden. $\triangleleft$

Eigenschaften von NFAs

Beobachtung 5

Seien $N_i = (Z_i, \Sigma, \delta_i, Q_i, E_i)$ NFAs mit $L(N_i) = L_i$ für $i = 1, 2$. Dann wird auch das Produkt $L_1 L_2$ von einem NFA erkannt.

Beweis

- Wir können $Z_1 \cap Z_2 = \emptyset$ annehmen.
- Die Sprache $L_1 L_2$ wird dann von dem NFA $N = (Z_1 \cup Z_2, \Sigma, \delta, Q_1, E)$ mit

$$\delta(p, a) = \begin{cases} \delta_1(p, a), & p \in Z_1 \setminus E_1, \\ \delta_1(p, a) \cup \bigcup_{q \in Q_2} \delta_2(q, a), & p \in E_1, \\ \delta_2(p, a), & \text{sonst} \end{cases}$$

und

$$E = \begin{cases} E_1 \cup E_2, & Q_2 \cap E_2 \neq \emptyset \\ E_2, & \text{sonst} \end{cases}$$

akzeptiert. □

Eigenschaften von NFAs

Beobachtung 6

Ist $N = (Z, \Sigma, \delta, Q_0, E)$ ein NFA, so wird auch die Sprache $L(N)^*$ von einem NFA erkannt.

Beweis

Die Sprache $L(N)^*$ wird von dem NFA

$$N^* = (Z \cup \{q_{neu}\}, \Sigma, \delta^*, Q_0 \cup \{q_{neu}\}, E \cup \{q_{neu}\})$$

mit

$$\delta^*(p, a) = \begin{cases} \delta(p, a) \cup \bigcup_{q \in Q_0} \delta(q, a), & p \in E, \\ \delta(p, a), & \text{sonst} \end{cases}$$

erkannt. $\square$

Überblick

Ziel

Zeige, dass REG als Abschluss der endl. Sprachen unter Vereinigung, Produktbildung und Sternhülle charakterisierbar ist.

Problem

Bei der Konstruktion eines DFA für das Produkt $L_1 L_2$ bereitet es Schwierigkeiten, den richtigen Zeitpunkt für den Übergang von (der Simulation von) M_1 zu M_2 zu finden.

Lösungsidee (bereits umgesetzt)

Ein nichtdeterministischer Automat (NFA) kann den richtigen Zeitpunkt für den Übergang „erraten“.

Noch zu zeigen

NFAs erkennen genau die regulären Sprachen.

NFAs erkennen genau die regulären Sprachen

Satz (Rabin und Scott)

$\mathrm{REG} = \{L(N) \mid N \text{ ist ein NFA}\}$.

Beweis von $\mathrm{REG} \subseteq \{L(N) \mid N \text{ ist ein NFA}\}$

Diese Inklusion ist klar, da jeder DFA $M = (Z, \Sigma, \delta, q_0, E)$ leicht in einen äquivalenten NFA

$$N = (Z, \Sigma, \bar{\delta}, Q_0, E)$$

transformiert werden kann, indem wir $\bar{\delta}(q, a) = \{\delta(q, a)\}$ und $Q_0 = \{q_0\}$ setzen. $\square$

Für die umgekehrte Inklusion ist das Erreichbarkeitsproblem für NFAs von zentraler Bedeutung.

Das Erreichbarkeitsproblem für NFAs

Frage

Sei $N = (Z, \Sigma, \delta, Q_0, E)$ ein NFA und sei $x = x_1 \cdots x_n \in \Sigma^*$. Welche Zustände kann $N(x)$ in i Schritten erreichen?

Antwort

- in 0 Schritten: alle Zustände in Q_0.
- in einem Schritt: alle Zustände in

$$Q_1 = \bigcup_{q \in Q_0} \delta(q, x_1).$$

- in i Schritten: alle Zustände in

$$Q_i = \bigcup_{q \in Q_{i-1}} \delta(q, x_i).$$

Simulation von NFAs durch DFAs

Idee

- Wir können einen NFA $N = (Z, \Sigma, \delta, Q_0, E)$ durch einen DFA M simulieren, der in seinem Zustand die Information speichert, in welchen Zuständen sich N momentan befinden könnte.
- Die Zustände von M sind also Teilmengen Q von Z mit Q_0 als Startzustand und der Endzustandsmenge

  $$E' = \{Q \subseteq Z \mid Q \cap E \neq \emptyset\}.$$

- Die Überführungsfunktion $\delta' : \mathcal{P}(Z) \times \Sigma \to \mathcal{P}(Z)$ von M berechnet dann für einen Zustand $Q \subseteq Z$ und ein Zeichen $a \in \Sigma$ die Menge

  $$\delta'(Q, a) = \bigcup_{q \in Q} \delta(q, a)$$

  aller Zustände, in die N gelangen kann, wenn N ausgehend von einem beliebigen Zustand $q \in Q$ das Zeichen a liest.
- Die von $M = (\mathcal{P}(Z), \Sigma, \delta', Q_0, E')$ erkannte Sprache ist

  $$L(M) = \{x \in \Sigma^* \mid \hat{\delta}'(Q_0, x) \cap E \neq \emptyset\}.$$

NFAs erkennen genau die regulären Sprachen

Beweis von $\{L(N) \mid N \text{ ist ein NFA}\} \subseteq \mathrm{REG}$

- Sei $N = (Z, \Sigma, \delta, Q_0, E)$ ein NFA und sei $M = (\mathcal{P}(Z), \Sigma, \delta', Q_0, E')$ der zugehörige Potenzmengenautomat mit $\delta'(Q, a) = \bigcup_{q \in Q} \delta(q, a)$ und $E' = \{Q \subseteq Z \mid Q \cap E \neq \emptyset\}$.
- Dann folgt die Korrektheit von M leicht mittels folgender Behauptung, deren Beweis wir auf der nächsten Folie nachholen.

Behauptung

$\hat{\delta}'(Q_0, x)$ enthält genau die von N nach Lesen von x erreichbaren Zustände.

- Für alle Wörter $x \in \Sigma^*$ gilt

$$\begin{aligned}
x \in L(N) \quad &\Leftrightarrow \quad N \text{ kann nach Lesen von } x \text{ einen Endzustand erreichen} \\
&\overset{\text{Beh.}}{\Leftrightarrow} \quad \hat{\delta}'(Q_0, x) \cap E \neq \emptyset \\
&\Leftrightarrow \quad \hat{\delta}'(Q_0, x) \in E' \\
&\Leftrightarrow \quad x \in L(M).
\end{aligned}$$

$\square$

Beweis der Behauptung

Behauptung

$\hat{\delta}'(Q_0, x)$ enthält genau die von N nach Lesen von x erreichbaren Zustände.

Beweis durch Induktion über die Länge n von x

$n = 0$: klar, da $\hat{\delta}'(Q_0, \varepsilon) = Q_0$ ist.

$n - 1 \rightsquigarrow n$: Sei $x = x_1 \cdots x_n$ gegeben. Nach IV enthält

$$Q_{n-1} = \hat{\delta}'(Q_0, x_1 \cdots x_{n-1})$$

die Zustände, die N nach Lesen von $x_1 \cdots x_{n-1}$ erreichen kann. Wegen

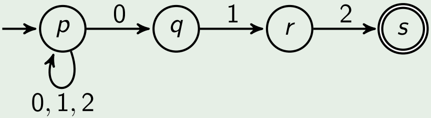
$$\hat{\delta}'(Q_0, x) = \delta'(Q_{n-1}, x_n) = \bigcup_{q \in Q_{n-1}} \delta(q, x_n)$$

enthält dann aber $\hat{\delta}'(Q_0, x)$ die Zustände, die N nach Lesen von x erreichen kann. $\square$

Simulation von NFAs durch DFAs

Beispiel

- Betrachte den NFA N



  mit Startzustandsmenge $Q_0 = \{p\}$ und Endzustandsmenge $E = \{s\}$.
- Ausgehend von Q_0 liefert δ' dann die folgenden Werte:

δ'	0	1	2
$\{p\}$	$\{p, q\}$	$\{p\}$	$\{p\}$
$\{p, q\}$	$\{p, q\}$	$\{p, r\}$	$\{p\}$
$\{p, r\}$	$\{p, q\}$	$\{p\}$	$\{p, s\}$
$\{p, s\}$	$\{p, q\}$	$\{p\}$	$\{p\}$

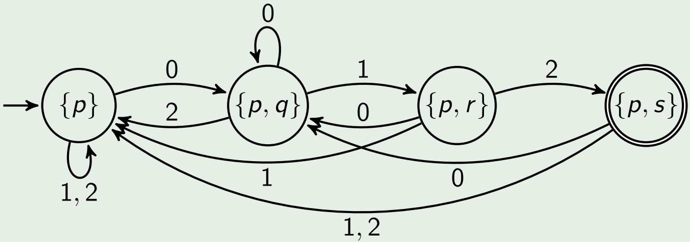
Simulation von NFAs durch DFAs

Beispiel

- Ausgehend von Q_0 liefert δ' dann die folgenden Werte:

δ'	0	1	2
$\{p\}$	$\{p, q\}$	$\{p\}$	$\{p\}$
$\{p, q\}$	$\{p, q\}$	$\{p, r\}$	$\{p\}$
$\{p, r\}$	$\{p, q\}$	$\{p\}$	$\{p, s\}$
$\{p, s\}$	$\{p, q\}$	$\{p\}$	$\{p\}$

- Also ist N äquivalent zu folgendem Potenzmengenautomaten M:



◁

Simulation von NFAs durch DFAs

Bemerkung

- Im obigen Beispiel werden für die Konstruktion des Potenzmengenautomaten nur 4 der insgesamt

  $$\|\mathcal{P}(Z)\| = 2^{\|Z\|} = 2^4 = 16$$

  Zustände benötigt, da die übrigen 12 Zustände nicht erreichbar sind.
- Es gibt jedoch Beispiele, bei denen alle $2^{\|Z\|}$ Zustände benötigt werden (siehe Übungen).

Abschlusseigenschaften der Klasse REG

Korollar

Die Klasse REG der regulären Sprachen ist unter folgenden Operationen abgeschlossen:

- Komplement,
- Durchschnitt,
- Vereinigung,
- Produkt,
- Sternhülle.

Überblick

Nächstes Ziel

Zeige, dass REG als Abschluss der endl. Sprachen unter Vereinigung, Produkt und Sternhülle charakterisierbar ist.

Bereits gezeigt:

Jede Sprache, die mittels der Operationen Vereinigung, Produkt und Sternhülle (sowie Durchschnitt und Komplement) angewandt auf endliche Sprachen darstellbar ist, ist regulär.

Noch zu zeigen:

Jede reguläre Sprache lässt sich aus endlichen Sprachen mittels Vereinigung, Produkt und Sternhülle erzeugen.

Konstruktive Charakterisierung von REG mittels regulärer Ausdrücke

Induktive Definition der Menge *RA* aller regulären Ausdrücke

Die Symbole $\emptyset$, ϵ und a ($a \in \Sigma$) sind reguläre Ausdrücke, die

- die leere Sprache $L(\emptyset) = \emptyset$,
- die Sprache $L(\epsilon) = \{\varepsilon\}$ und
- für jedes $a \in \Sigma$ die Sprache $L(a) = \{a\}$ beschreiben.

Sind α und β reguläre Ausdrücke, die die Sprachen $L(\alpha)$ und $L(\beta)$ beschreiben, so sind auch $\alpha\beta$, $(\alpha|\beta)$ und $(\alpha)^*$ reguläre Ausdrücke, die folgende Sprachen beschreiben:

- $L(\alpha\beta) = L(\alpha)L(\beta)$,
- $L(\alpha|\beta) = L(\alpha) \cup L(\beta)$,
- $L((\alpha)^*) = L(\alpha)^*$.

Reguläre Ausdrücke

Beispiel

Die regulären Ausdrücke ϵ^*, $\emptyset^*$, $(0|1)^*00$ und $(\epsilon 0|\emptyset 1^*)$ beschreiben folgende Sprachen:

γ	ϵ^*	$\emptyset^*$	$(0\|1)^*00$	$(\epsilon 0\|\emptyset 1^*)$
$L(\gamma)$	$\{\varepsilon\}^* = \{\varepsilon\}$	$\emptyset^* = \{\varepsilon\}$	$\{x00 \mid x \in \{0,1\}^*\}$	$\{0\}$

◁

Vereinbarungen

- Um Klammern zu sparen, definieren wir folgende Präzedenzordnung: Der Sternoperator * bindet stärker als der Produktoperater und dieser wiederum stärker als der Vereinigungsoperator.
- Für $((a|b(c)^*)|d)$ können wir also kurz $a|bc^*|d$ schreiben.
- Da der reguläre Ausdruck $\gamma\gamma^*$ die Sprache $L(\gamma)^+$ beschreibt, verwenden wir γ^+ als Abkürzung für den Ausdruck $\gamma\gamma^*$.

Charakterisierung von REG durch reg. Ausdrücke

Satz

$\{L(\gamma) \mid \gamma$ ist ein regulärer Ausdruck$\} \subseteq$ REG.

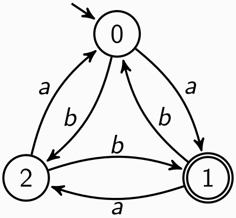
Beweis.

Klar, da

- die Basisausdrücke $\emptyset$, ϵ und a, $a \in \Sigma^*$, nur reguläre Sprachen beschreiben und
- die Sprachklasse REG unter Produkt, Vereinigung und Sternhülle abgeschlossen ist. $\square$

Charakterisierung von REG durch reg. Ausdrücke

M_3:



Frage

Wie lässt sich die Sprache

$$L(M_3) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}$$

durch einen regulären Ausdruck beschreiben?

Antwort

- Die Sprache $L_0 = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 0\}$ lässt sich durch folgenden regulären Ausdruck beschreiben:

$$\gamma_0 = (a(ab)^*(aa|b) \mid b(ba)^*(a|bb))^*.$$

- Also ist $L(M_3)$ durch folgenden regulären Ausdruck beschreibbar:

$$\gamma_1 = \gamma_0(a|bb)(ab)^*.$$

Charakterisierung von REG durch reg. Ausdrücke

Satz

REG $\subseteq \{L(\gamma) \mid \gamma$ ist ein regulärer Ausdruck$\}$.

Beweis

- Wir konstruieren zu einem DFA $M = (Z, \Sigma, \delta, q_0, E)$ einen regulären Ausdruck γ mit $L(\gamma) = L(M)$.
- Wir nehmen an, dass $Z = \{1, \ldots, m\}$ und $q_0 = 1$ ist.
- Dann lässt sich $L(M)$ als Vereinigung

  $$L(M) = \bigcup_{q \in E} L_{1,q}$$

  von Sprachen der Form $L_{p,q} = \{x \in \Sigma^* \mid \hat{\delta}(p, x) = q\}$ darstellen.
- Es reicht also, reguläre Ausdrücke für die Sprachen $L_{p,q}$ mit $1 \leq p, q \leq m$ anzugeben.

Charakterisierung von REG durch reg. Ausdrücke

Satz

REG $\subseteq \{L(\gamma) \mid \gamma$ ist ein regulärer Ausdruck$\}$.

Beweis (Fortsetzung)

- Es reicht also, reguläre Ausdrücke für die Sprachen $L_{p,q}$ mit $1 \leq p, q \leq m$ anzugeben.
- Hierzu betrachten wir für $r = 0, \ldots, m$ die Sprachen

$$L^r_{p,q} = \left\{ x \in L_{p,q} \,\middle|\, \text{für } i = 1, \ldots, n-1 \text{ ist } \hat{\delta}(p, x_1 \cdots x_i) \leq r \right\}.$$

- Wegen $L_{p,q} = L^m_{p,q}$ reicht es, reguläre Ausdrücke für die Sprachen $L^r_{p,q}$ mit $1 \leq p, q \leq m$ und $0 \leq r \leq m$ anzugeben.
- Wir zeigen induktiv über r, dass die Sprachen $L^r_{p,q}$ durch reguläre Ausdrücke beschreibbar sind.

Charakterisierung von REG durch reg. Ausdrücke

Satz

REG $\subseteq \{L(\gamma) \mid \gamma$ ist ein regulärer Ausdruck$\}$.

Beweis (Schluss)

- Wir zeigen induktiv über r, dass die Sprachen $L^r_{p,q}$ durch reguläre Ausdrücke beschreibbar sind.

  $r = 0$: In diesem Fall sind die Sprachen

  $$L^0_{p,q} = \begin{cases} \{a \in \Sigma \mid \delta(p,a) = q\}, & p \neq q, \\ \{a \in \Sigma \mid \delta(p,a) = q\} \cup \{\varepsilon\}, & \text{sonst} \end{cases}$$

  endlich und somit durch reguläre Ausdrücke beschreibbar.

  $r \rightsquigarrow r+1$: Wegen

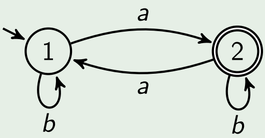
  $$L^{r+1}_{p,q} = L^r_{p,q} \cup L^r_{p,r+1}(L^r_{r+1,r+1})^* L^r_{r+1,q}$$

  sind mit $L^r_{p,q}$, $1 \leq p,q \leq m$, auch die Sprachen $L^{r+1}_{p,q}$, $1 \leq p,q \leq m$, durch reguläre Ausdrücke beschreibbar. $\square$

Charakterisierung von REG durch reg. Ausdrücke

Beispiel

- Betrachte den DFA M



- Da M insgesamt $m = 2$ Zustände und nur den Endzustand 2 besitzt, ist

$$L(M) = \bigcup_{q \in E} L_{1,q} = L_{1,2} = L_{1,2}^{2}.$$

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

- Um reguläre Ausdrücke $\gamma^r_{p,q}$ für die Sprachen $L^r_{p,q}$ zu bestimmen, benutzen wir für $r \geq 0$ die Rekursionsformel

$$\gamma^{r+1}_{p,q} = \gamma^r_{p,q} | \gamma^r_{p,r+1}(\gamma^r_{r+1,r+1})^* \gamma^r_{r+1,q}.$$

- Damit erhalten wir

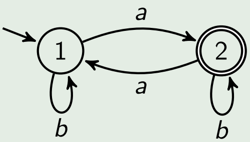
$$\begin{aligned}
\gamma^2_{1,2} &= \gamma^1_{1,2} | \gamma^1_{1,2}(\gamma^1_{2,2})^* \gamma^1_{2,2}, \\
\gamma^1_{1,2} &= \gamma^0_{1,2} | \gamma^0_{1,1}(\gamma^0_{1,1})^* \gamma^0_{1,2}, \\
\gamma^1_{2,2} &= \gamma^0_{2,2} | \gamma^0_{2,1}(\gamma^0_{1,1})^* \gamma^0_{1,2}.
\end{aligned}$$

- Es genügt also, die regulären Ausdrücke $\gamma^0_{1,1}$, $\gamma^0_{1,2}$, $\gamma^0_{2,1}$, $\gamma^0_{2,2}$, $\gamma^1_{1,2}$, $\gamma^1_{2,2}$ und $\gamma^2_{1,2}$ zu berechnen.

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformeln

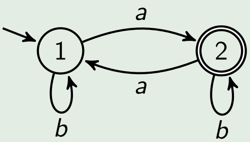
$$L^0_{p,p} = \{a \mid \delta(p,a) = p\} \cup \{\varepsilon\},$$
$$L^0_{p,q} = \{a \mid \delta(p,a) = q\} \text{ für } p \neq q,$$
$$L^{r+1}_{p,q} = L^r_{p,q} \cup L^r_{p,r+1}(L^r_{r+1,r+1})^* L^r_{r+1,q}.$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r=0$				
$r=1$				
$r=2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

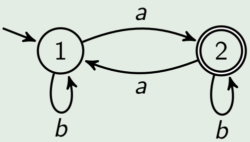
$$L^0_{1,1} = \{a \in \Sigma \mid \delta(1,a) = 1\} \cup \{\varepsilon\} = \{\varepsilon, b\}$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r=0$				
$r=1$				
$r=2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

$$L^0_{1,1} = \{a \in \Sigma \mid \delta(1, a) = 1\} \cup \{\varepsilon\} = \{\varepsilon, b\}$$

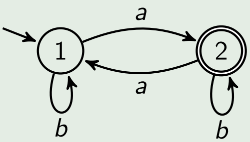
$$\rightsquigarrow \quad \gamma^0_{1,1} = \epsilon | b$$

(p, q)	$(1, 1)$	$(1, 2)$	$(2, 1)$	$(2, 2)$
$r = 0$	$\epsilon \vert b$			
$r = 1$				
$r = 2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

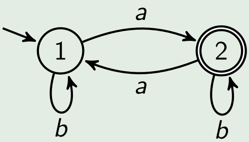
$$L^0_{1,2} = \{a \in \Sigma \mid \delta(1,a) = 2\} = \{a\}$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r=0$	$\epsilon \mid b$			
$r=1$				
$r=2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

$$L^0_{1,2} = \{a \in \Sigma \mid \delta(1,a) = 2\} = \{a\}$$

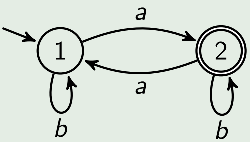
$$\rightsquigarrow \quad \gamma^0_{1,2} = a$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r=0$	$\epsilon \mid b$	a		
$r=1$				
$r=2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

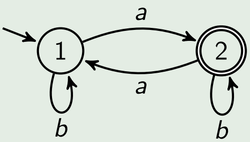
$$L^0_{2,1} = \{a \in \Sigma \mid \delta(2,a) = 1\} = \{a\}$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r=0$	$\epsilon \vert b$	a		
$r=1$				
$r=2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

$$L^0_{2,1} = \{a \in \Sigma \mid \delta(2, a) = 1\} = \{a\}$$

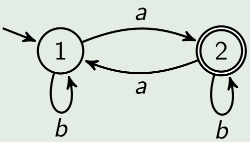
$$\rightsquigarrow \quad \gamma^0_{2,1} = a$$

(p, q)	$(1, 1)$	$(1, 2)$	$(2, 1)$	$(2, 2)$
$r = 0$	$\epsilon \vert b$	a	a	
$r = 1$				
$r = 2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

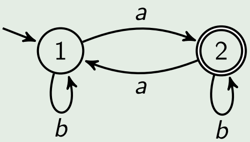
$$L_{2,2}^{0} = \{a \in \Sigma \mid \delta(2, a) = 2\} \cup \{\varepsilon\} = \{\varepsilon, b\}$$

(p, q)	$(1, 1)$	$(1, 2)$	$(2, 1)$	$(2, 2)$
$r = 0$	$\epsilon \mid b$	a	a	
$r = 1$				
$r = 2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

$$L^0_{2,2} = \{a \in \Sigma \mid \delta(2, a) = 2\} \cup \{\varepsilon\} = \{\varepsilon, b\}$$

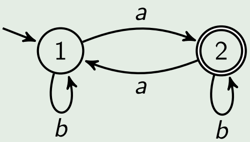
$$\rightsquigarrow \quad \gamma^0_{2,2} = \epsilon | b$$

(p, q)	$(1, 1)$	$(1, 2)$	$(2, 1)$	$(2, 2)$
$r = 0$	$\epsilon \| b$	a	a	$\epsilon \| b$
$r = 1$	-			
$r = 2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

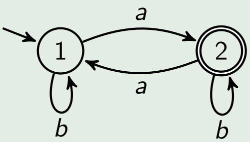
$$\gamma^1_{1,2} = \gamma^0_{1,2} | \gamma^0_{1,1} (\gamma^0_{1,1})^* \gamma^0_{1,2}$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r=0$	$\epsilon \mid b$	a	a	$\epsilon \mid b$
$r=1$	-			
$r=2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

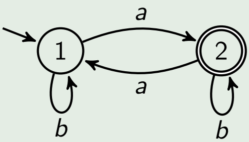
$$\begin{aligned}\gamma_{1,2}^{1} &= \gamma_{1,2}^{0} | \gamma_{1,1}^{0} (\gamma_{1,1}^{0})^{*} \gamma_{1,2}^{0} \\ &= a | (\epsilon | b)(\epsilon | b)^{*} a\end{aligned}$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r=0$	$\epsilon \| b$	a	a	$\epsilon \| b$
$r=1$	-			
$r=2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

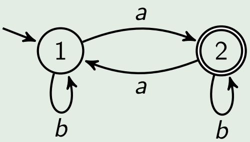
$$\begin{aligned}\gamma^{1}_{1,2} &= \gamma^{0}_{1,2} | \gamma^{0}_{1,1} (\gamma^{0}_{1,1})^{*} \gamma^{0}_{1,2} \\ &= a | (\epsilon | b)(\epsilon | b)^{*} a \\ &\equiv b^{*} a\end{aligned}$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r=0$	$\epsilon \| b$	a	a	$\epsilon \| b$
$r=1$	-	$b^{*}a$	-	
$r=2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

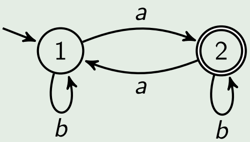
$$\gamma_{2,2}^{1} = \gamma_{2,2}^{0} | \gamma_{2,1}^{0} (\gamma_{1,1}^{0})^{*} \gamma_{1,2}^{0}$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r=0$	$\epsilon \vert b$	a	a	$\epsilon \vert b$
$r=1$	-	$b^{*}a$	-	
$r=2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

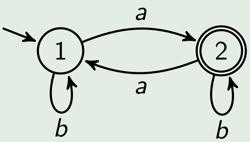
$$\gamma^1_{2,2} = \gamma^0_{2,2} | \gamma^0_{2,1} (\gamma^0_{1,1})^* \gamma^0_{1,2}$$
$$= (\epsilon | b) | a (\epsilon | b)^* a$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r = 0$	$\epsilon \vert b$	a	a	$\epsilon \vert b$
$r = 1$	-	$b^* a$	-	
$r = 2$				

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

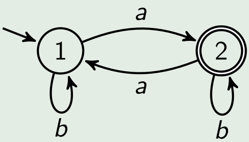
$$
\begin{aligned}
\gamma_{2,2}^{1} &= \gamma_{2,2}^{0} | \gamma_{2,1}^{0} (\gamma_{1,1}^{0})^* \gamma_{1,2}^{0} \\
&= (\epsilon|b)|a(\epsilon|b)^* a \\
&\equiv \epsilon|b|ab^*a
\end{aligned}
$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r=0$	$\epsilon\|b$	a	a	$\epsilon\|b$
$r=1$	-	b^*a	-	$\epsilon\|b\|ab^*a$
$r=2$	-			

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformeln

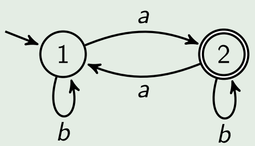
$$\gamma_{1,2}^{2} = \gamma_{1,2}^{1} | \gamma_{1,2}^{1} (\gamma_{2,2}^{1})^{*} \gamma_{2,2}^{1}$$

(p, q)	$(1, 1)$	$(1, 2)$	$(2, 1)$	$(2, 2)$
$r = 0$	$\epsilon \| b$	a	a	$\epsilon \| b$
$r = 1$	-	$b^{*}a$	-	$\epsilon \| b \| ab^{*}a$
$r = 2$	-			

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

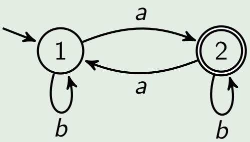
$$\gamma^2_{1,2} = \gamma^1_{1,2} | \gamma^1_{1,2} (\gamma^1_{2,2})^* \gamma^1_{2,2}$$
$$= b^*a | b^*a(\epsilon|b|ab^*a)^*(\epsilon|b|ab^*a)$$

(p,q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r=0$	$\epsilon \| b$	a	a	$\epsilon \| b$
$r=1$	-	b^*a	-	$\epsilon \| b \| ab^*a$
$r=2$	-			

Charakterisierung von REG durch reg. Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

$$
\begin{aligned}
\gamma_{1,2}^2 &= \gamma_{1,2}^1 | \gamma_{1,2}^1 (\gamma_{2,2}^1)^* \gamma_{2,2}^1 \\
&= b^* a | b^* a (\epsilon | b | ab^* a)^* (\epsilon | b | ab^* a) \\
&\equiv b^* a (b | ab^* a)^*
\end{aligned}
$$

(p, q)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$r = 0$	$\epsilon \mid b$	a	a	$\epsilon \mid b$
$r = 1$	-	$b^* a$	-	$\epsilon \mid b \mid ab^* a$
$r = 2$	-	$b^* a (b \mid ab^* a)^*$	-	-

◁

Charakterisierungen der Klasse REG

Korollar

Sei L eine Sprache. Dann sind folgende Aussagen äquivalent:

- L ist regulär,
- es gibt einen DFA M mit $L = L(M)$,
- es gibt einen NFA N mit $L = L(N)$,
- es gibt einen regulären Ausdruck γ mit $L = L(\gamma)$,
- L lässt sich mit den Operationen Vereinigung, Produkt und Sternhülle aus endlichen Sprachen gewinnen,
- L lässt sich mit den Operationen $\cap$, $\cup$, Komplement, Produkt und Sternhülle aus endlichen Sprachen gewinnen.

Ausblick

- Als nächstes wenden wir uns der Frage zu, wie sich die Anzahl der Zustände eines DFA minimieren lässt.
- Da hierbei Äquivalenzrelationen eine wichtige Rolle spielen, befassen wir uns zunächst mit Relationalstrukturen.

Relationalstrukturen

Definition

- Sei A eine nichtleere Menge, R ist eine k-stellige Relation auf A, wenn $R \subseteq A^k = \underbrace{A \times \cdots \times A}_{k\text{-mal}}$ ist.
- Für $i = 1, \ldots, n$ sei R_i eine k_i-stellige Relation auf A. Dann heißt $(A; R_1, \ldots, R_n)$ Relationalstruktur.
- Die Menge A heißt der Individuenbereich, die Trägermenge oder die Grundmenge der Relationalstruktur.

Bemerkung

- Wir werden hier hauptsächlich den Fall $n = 1$, $k_1 = 2$, also (A, R) mit $R \subseteq A \times A$ betrachten.
- Man nennt dann R eine (binäre) Relation auf A.
- Oft wird für $(a, b) \in R$ auch die Infix-Schreibweise aRb benutzt.

Relationalstrukturen

Beispiel

- (F, M) mit $F = \{f \mid f \text{ ist Fluss in Europa}\}$ und
  $$M = \{(f, g) \in F \times F \mid f \text{ mündet in } g\},$$
- (U, B) mit $U = \{x \mid x \text{ ist Berliner }\}$ und
  $$B = \{(x, y) \in U \times U \mid x \text{ ist Bruder von } y\},$$
- $(\mathcal{P}(M), \subseteq)$, wobei M eine beliebige Menge und $\subseteq$ die Inklusionsrelation auf den Teilmengen von M ist,
- (A, Id_A) mit $Id_A = \{(x, x) \mid x \in A\}$ (die Identität auf A),
- $(\mathbb{R}, \leq)$,
- $(\mathbb{Z}, |)$, wobei $|$ die "teilt"-Relation bezeichnet,
- $(Fml, \Rightarrow)$ mit $Fml = \{F \mid F \text{ ist aussagenlogische Formel}\}$ und
  $$\Rightarrow \; = \; \{(F, G) \in Fml \times Fml \mid G \text{ ist log. Folgerung von } F\}.$$

◁

Mengentheoretische Operationen auf Relationen

- Da Relationen Mengen sind, können wir den Durchschnitt, die Vereinigung, die Differenz und das Komplement von Relationen bilden:

$$
\begin{aligned}
R \cap S &= \{(x, y) \in A \times A \mid xRy \wedge xSy\}, \\
R \cup S &= \{(x, y) \in A \times A \mid xRy \vee xSy\}, \\
R - S &= \{(x, y) \in A \times A \mid xRy \wedge \neg xSy\}, \\
\overline{R} &= (A \times A) - R.
\end{aligned}
$$

- Sei $\mathcal{M} \subseteq \mathcal{P}(A \times A)$ eine beliebige Menge von Relationen auf A. Dann sind der Schnitt über $\mathcal{M}$ und die Vereinigung über $\mathcal{M}$ folgende Relationen:

$$
\begin{aligned}
\bigcap \mathcal{M} &= \{(x, y) \mid \forall R \in \mathcal{M} : xRy\}, \\
\bigcup \mathcal{M} &= \{(x, y) \mid \exists R \in \mathcal{M} : xRy\}.
\end{aligned}
$$

- Weiterhin ist die Inklusion $R \subseteq S$ auf Relationen definiert. Es gilt

$$R \subseteq S \Leftrightarrow \forall x, y : xRy \rightarrow xSy.$$

Weitere Operationen auf Relationen

Definition

- Die transponierte (konverse) Relation zu R ist
$$R^T = \{(y, x) \mid xRy\}.$$
- R^T wird oft auch mit R^{-1} bezeichnet.
- Zum Beispiel ist $(\mathbb{R}, \leq^T) = (\mathbb{R}, \geq)$.
- Das Produkt (oder die Komposition) zweier Relationen R und S ist
$$R \circ S = \{(x, z) \in A \times A \mid \exists y \in A : xRy \wedge ySz\}.$$

Beispiel

Ist B die Relation "ist Bruder von", V "ist Vater von", M "ist Mutter von" und $E = V \cup M$ "ist Elternteil von", so ist $B \circ E$ die Onkel-Relation. $\triangleleft$

Das Relationenprodukt

Notation

- Für $R \circ S$ wird auch $R\,;S$, $R \cdot S$ oder einfach RS geschrieben.
- Für $\underbrace{R \circ \cdots \circ R}_{\text{n-mal}}$ schreiben wir auch R^n. Dabei ist $R^0 = Id$.

Vorsicht!

Das Relationenprodukt R^n sollte nicht mit dem kartesischen Produkt

$$\underbrace{R \times \cdots \times R}_{\text{n-mal}}$$

verwechselt werden.

Vereinbarung

Wir vereinbaren, dass R^n das n-fache Relationenprodukt bezeichnen soll, falls R eine Relation ist.

Eigenschaften von Relationen

Definition

Sei R eine Relation auf A. Dann heißt R

reflexiv,	falls $\forall x \in A : xRx$	(also $Id_A \subseteq R$)
irreflexiv,	falls $\forall x \in A : \neg xRx$	(also $Id_A \subseteq \overline{R}$)
symmetrisch,	falls $\forall x, y \in A : xRy \Rightarrow yRx$	(also $R \subseteq R^T$)
asymmetrisch,	falls $\forall x, y \in A : xRy \Rightarrow \neg yRx$	(also $R \subseteq \overline{R^T}$)
antisymmetrisch,	falls $\forall x, y \in A : xRy \wedge yRx \Rightarrow x = y$	(also $R \cap R^T \subseteq Id$)
konnex,	falls $\forall x, y \in A : xRy \vee yRx$	(also $A \times A \subseteq R \cup R^T$)
semikonnex,	falls $\forall x, y \in A : x \neq y \Rightarrow xRy \vee yRx$	(also $\overline{Id} \subseteq R \cup R^T$)
transitiv,	falls $\forall x, y, z \in A : xRy \wedge yRz \Rightarrow xRz$	(also $R^2 \subseteq R$)

gilt.

Eigenschaften von Relationen

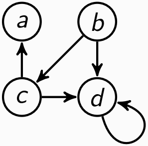
Beispiel

- Die Relation "ist Schwester von" ist zwar in einer reinen Damengesellschaft symmetrisch, i.a. jedoch weder symmetrisch noch asymmetrisch noch antisymmetrisch.
- $(\mathbb{R}, <)$ ist irreflexiv, asymmetrisch, transitiv und semikonnex.
- $(\mathbb{R}, \leq)$ und $(\mathcal{P}(M), \subseteq)$ sind reflexiv, antisymmetrisch und transitiv.
- $(\mathbb{R}, \leq)$ ist auch konnex.
- $(\mathcal{P}(M), \subseteq)$ ist zwar im Fall $\|M\| \leq 1$ konnex, aber im Fall $\|M\| \geq 2$ weder semikonnex noch konnex. $\triangleleft$

Darstellung von endlichen Relationen

Graphische Darstellung

$$A = \{a, b, c, d\}$$
$$R = \{(b, c), (b, d), (c, a), (c, d), (d, d)\}$$



- Eine Relation R auf einer endlichen Menge A kann durch einen gerichteten Graphen (kurz Digraphen) $G = (A, R)$ mit Knotenmenge A und Kantenmenge R veranschaulicht werden.
- Hierzu stellen wir jedes Element $x \in A$ als einen Knoten dar und verbinden jedes Knotenpaar $(x, y) \in R$ durch eine gerichtete Kante (Pfeil).
- Zwei durch eine Kante verbundene Knoten heißen adjazent oder benachbart.

Darstellung von endlichen Relationen

Definition

Sei R eine binäre Relation auf A.

- Der Nachbereich von x ist

$$R(x) = \{y \in A \mid xRy\}.$$

- Der Ausgangsgrad eines Knotens x ist

$$\deg^+(x) = \|R(x)\|.$$

- Der Eingangsgrad von x ist

$$\deg^-(x) = \|\{y \in A \mid yRx\}\| = \|R^{-1}(x)\|.$$

- Ist R symmetrisch, so können wir die Pfeilspitzen auch weglassen.
- In diesem Fall ist $\deg(x) = \deg^-(x) = \deg^+(x)$ der Grad von x in G.
- G ist schleifenfrei, falls R irreflexiv ist.
- Ist R irreflexiv und symmetrisch, so nennen wir G einen (ungerichteten) Graphen.

Darstellung von endlichen Relationen

Matrixdarstellung (Adjazenzmatrix)

Eine Relation R auf $A = \{a_1, \ldots, a_n\}$ lässt sich auch durch die boolesche $(n \times n)$-Matrix $M_R = (m_{ij})$ darstellen mit

$$m_{ij} = \begin{cases} 1, & a_i R a_j, \\ 0, & \text{sonst.} \end{cases}$$

Beispiel

Die Relation $R = \{(b,c), (b,d), (c,a), (c,d), (d,d)\}$ auf $A = \{a,b,c,d\}$ hat beispielsweise die Matrixdarstellung

$$M_R = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

◁

Darstellung von endlichen Relationen

Tabellendarstellung (Adjazenzliste)

R lässt sich auch durch eine Tabelle darzustellen, die jedem Element $x \in A$ seinen Nachbereich $R(x)$ in Form einer Liste zuordnet.

Beispiel

Die Relation $R = \{(b, c), (b, d), (c, a), (c, d), (d, d)\}$ auf $A = \{a, b, c, d\}$ hat beispielsweise die Tabellendarstellung

x	$R(x)$
a	-
b	c, d
c	a, d
d	d

$\triangleleft$

Berechnung des Relationenprodukts

Berechnung von $R \circ S$

- Sind $M_R = (r_{ij})$ und $M_S = (s_{ij})$ boolesche $n \times n$-Matrizen für R und S, so erhalten wir für $T = R \circ S$ die Matrix $M_T = (t_{ij})$ mit

$$t_{ij} = \bigvee_{k=1,\ldots,n} (r_{ik} \wedge s_{kj}).$$

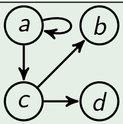
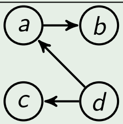
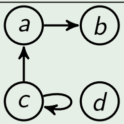
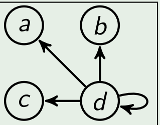
- Der Nachbereich $T(x)$ von x bzgl. der Relation $T = R \circ S$ berechnet sich zu

$$T(x) = \bigcup_{y \in R(x)} S(y).$$

Das Relationenprodukt

Beispiel

Betrachte die Relationen $R = \{(a,a),(a,c),(c,b),(c,d)\}$ und $S = \{(a,b),(d,a),(d,c)\}$ auf der Menge $A = \{a,b,c,d\}$.

	R	S	$R \circ S$	$S \circ R$
Digraph				
Adjazenz-matrix	$\begin{matrix}1&0&1&0\\0&0&0&0\\0&1&0&1\\0&0&0&0\end{matrix}$	$\begin{matrix}0&1&0&0\\0&0&0&0\\0&0&0&0\\1&0&1&0\end{matrix}$	$\begin{matrix}0&1&0&0\\0&0&0&0\\1&0&1&0\\0&0&0&0\end{matrix}$	$\begin{matrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\1&1&1&1\end{matrix}$
Adjazenz-liste	a : a, c b : - c : b, d d : -	a : b b : - c : - d : a, c	a : b b : - c : a, c d : -	a : - b : - c : - d : a, b, c, d

◁

Das Relationenprodukt

Beobachtung

Das Relationenprodukt ist nicht kommutativ, d.h. i.a. gilt nicht $R \circ S = S \circ R$.

Relationenalgebra

Als nächstes zeigen wir, dass die Menge $\mathcal{R} = \mathcal{P}(A \times A)$ aller binären Relationen auf A mit dem Relationenprodukt $\circ$ als binärer Operation und der Relation Id_A als neutralem Element eine Halbgruppe (oder Monoid) bildet.

Satz

Seien Q, R, S Relationen auf A. Dann gilt

1. $(Q \circ R) \circ S = Q \circ (R \circ S)$, d.h. $\circ$ ist assoziativ,
2. $Id \circ R = R \circ Id = R$, d.h. Id ist neutrales Element.

Relationenalgebra

Satz

Seien Q, R, S Relationen auf A. Dann gilt

1. $(Q \circ R) \circ S = Q \circ (R \circ S)$, d.h. $\circ$ ist assoziativ,
2. $Id \circ R = R \circ Id = R$, d.h. Id ist neutrales Element.

Beweis.

1. $$\begin{aligned} x\ (Q \circ R) \circ S\ y &\Leftrightarrow \exists u : x\ (Q \circ R)\ u \ \wedge\ u\ S\ y \\ &\Leftrightarrow \exists u : (\exists v : x\ Q\ v\ R\ u) \ \wedge\ u\ S\ y \\ &\Leftrightarrow \exists u, v : x\ Q\ v\ R\ u\ S\ y \\ &\Leftrightarrow \exists v : x\ Q\ v \ \wedge\ (\exists u : v\ R\ u \ \wedge\ u\ S\ y) \\ &\Leftrightarrow \exists v : x\ Q\ v\ (R \circ S)\ y \\ &\Leftrightarrow x\ Q \circ (R \circ S)\ y \end{aligned}$$
2. Wegen $x\ Id \circ R\ y \Leftrightarrow \exists z : x = z \ \wedge\ z\ R\ y \Leftrightarrow x\ R\ y$ folgt $Id \circ R = R$. Die Gleichheit $R \circ Id = R$ folgt analog. □

Hüllenoperatoren

Bemerkung

- Es ist leicht zu sehen, dass der Schnitt von transitiven Relationen ebenfalls transitiv ist.
- Die transitive Hülle von R ist

$$R^+ = \bigcap \{S \subseteq A \times A \mid S \text{ ist transitiv und } R \subseteq S\}.$$

- R^+ ist also eine transitive Relation, die R enthält.
- Da R^+ zudem in jeder Relation mit diesen Eigenschaften enthalten ist, ist R^+ die kleinste transitive Relation, die R enthält.
- Da auch die Reflexivität und die Symmetrie bei der Schnittbildung erhalten bleiben, lassen sich nach demselben Muster weitere Hüllenoperatoren definieren.

Weitere Hüllenoperatoren

Bemerkung

- Die reflexive Hülle von R ist
$$h_{\mathrm{refl}}(R) = \bigcap\{S \subseteq A \times A \mid S \text{ ist reflexiv und } R \subseteq S\}.$$
- Die symmetrische Hülle von R ist
$$h_{\mathrm{sym}}(R) = \bigcap\{S \subseteq A \times A \mid S \text{ ist symmetrisch und } R \subseteq S\}.$$
- Die reflexiv-transitive Hülle von R ist
$$R^* = \bigcap\{S \subseteq A \times A \mid S \text{ ist reflexiv, transitiv und } R \subseteq S\}.$$

Transitive und reflexive Hülle

Satz

$h_{\text{refl}}(R) = R \cup Id_A, \;\; h_{\text{sym}}(R) = R \cup R^T, \;\; R^+ = \bigcup_{n \geq 1} R^n, \;\; R^* = \bigcup_{n \geq 0} R^n.$

Beweis

Siehe Übungen. □

Bemerkung

- Ein Paar (a, b) ist also genau dann in der reflexiv-transitiven Hülle R^* von R enthalten, wenn es ein $n \geq 0$ gibt mit $aR^n b$.
- Dies bedeutet, dass es Elemente $x_0, \ldots, x_n \in A$ gibt mit

$$x_0 = a, \; x_n = b \;\text{ und }\; x_0 R x_1 R x_2 \cdots x_{n-1} R x_n.$$

- $x_0, \ldots, x_n$ heißt Weg der Länge n von a nach b.

Überblick über Relationalstrukturen

Äquivalenz- und Ordnungsrelationen

	refl.	sym.	trans.	antisym.	asym.	konnex	semikon.
Äquivalenzrelation	✓	✓	✓				
(Halb-)Ordnung	✓		✓	✓			
Striktordnung			✓		✓		
lineare Ordnung			✓	✓		✓	
lin. Striktord.			✓		✓		✓
Quasiordnung	✓		✓				

Bemerkung

In der Tabelle sind nur die definierenden Eigenschaften durch ein "✓" gekennzeichnet. Das schließt nicht aus, dass noch weitere Eigenschaften vorliegen.

Äquivalenzrelationen

Beispiel

- Auf der Menge aller Geraden im $\mathbb{R}^2$ die Parallelität.
- Auf der Menge aller Menschen "im gleichen Jahr geboren wie".
- Auf $\mathbb{Z}$ die Relation "gleicher Rest bei Division durch m".
- Auf der Menge aller aussagenlogischen Formeln die semantische Äquivalenz. $\triangleleft$